A Comparative Study of Uncertainty in Transformer Networks with Monte Carlo Dropout

Abstract—**In recent years, language models that can do human-like linguistic tasks have outperformed all expectations, and the transformer model is one of the major successes in natural language processing (NLP). It has significantly altered how we work with text data. Understanding the reliability and confidence of transformer model predictions is essential for constructing trustworthy machine learning systems. A quantitative and comparative analysis of uncertainty among different transformer architecture is still under-explored. In this study, we examine the uncertainty measurement for Transformer-based models such as BERT, and XLNet. We employ dropouts during the inference phase (Monte Carlo Dropout) to quantify the uncertainty of these transformer models. Monte Carlo Dropout (MCD) has negligible computation cost and aids in the separation of uncertain samples and predictions.**

Index Terms—**transformer, uncertainty, monte carlo dropout**

I. Introduction

Natural language processing (NLP) systems that are dependable, accountable, and trustworthy must quantify the uncertainty of their machine learning models. Obtaining measurements of uncertainty in predictions enables the identification of out-of-domain [1], adversarial, or error-prone occurrences requiring particular treatment. For instance, such occurrences may be subjected to additional review by human specialists or a more advanced technology, or they may be rejected from classification [2]. Moreover, uncertainty estimation is an integral part of numerous applications, including active learning [3] and outlier/error identification in a dataset (Larson et al., 2019). BERT [4] and ELECTRA [5], for example, take advantage of deep pre-trained models based on the Transformer architecture [6] ([4]; [5]. Consequently, obtaining accurate uncertainty estimates for such neural networks (NNs) can directly benefit a wide range of natural language processing tasks; yet, implementing uncertainty estimation in this situation is problematic due to the vast number of parameters in these deep learning models. The approximations of Bayesian inference based on dropout utilization during the inference stage - Monte Carlo Dropout (MCD) [7] offer a realizable method for estimating the uncertainty of deep models. Due to the necessity of performing several stochastic forecasts, they are typically accompanied by significant processing overhead. Importantly, training ensembles of independent models results in even more prohibitive overheads [8].

In this wor

II. Literature Review

Uncertainty can be measured in several approaches. Three of the most popular ways to estimate uncertainty are Dropout as a Bayesian Approximation (Monte Carlo Dropout) [7], ensembling, where a discrepancy between models' predictions are interpreted as a sample variance [8], and Bayesian neural networks [9]. MCD is the most convenient technique to build uncertainty-aware models in three of them, and it also has other advantages, such as minimizing overfitting, decreasing model complexity, etc.

Shelmanov et al. [3] compare multiple uncertain estimates in text classification tasks for the cutting-edge Transformer model ELECTRA and the speed-oriented DistilBERT model. They use many stochastic passes with the MCD and a dropout based on Determinantal Point Processes to derive uncertainty estimates. Hu et al. [10] also suggest using empirical uncertainty in out-of-distribution identification for tasks involving text classification. They present a low-cost framework that uses auxiliary outliers as well as pseudo off-manifold samples to train the model with previous knowledge of a certain class, which has a high vacuity for out-of-distribution data. Vazhentsev et al. [11] employ Diverse Determinantal Point Process Monte Carlo Dropout to measure uncertainty and provide two optimizations to transformer models that reduce computation time and improve misclassification detection in named entity recognition and text classification.

III. Uncertainty Estimation

A. Monte Carlo Dropout

Dropout is usually implemented to reduce model complexity as well as to avoid overfitting [12]. In the training stage, the output of every neuron is proliferated using a binary mask which is derived from Bernoulli distribution. This is how the neurons are initialized to zero, following which the model is applied at the testing stage. The idea of employing dropout was brought forth by Gal and Ghahramani [7], who used it as an estimation of probabilistic Bayesian models for deep Gaussian processes. An ensemble of predictions showcasing the uncertainty estimations can be generated using MCD. The MCD method involves executing several stochastic forward passes in a model by employing activated dropout during the testing stage.

If we are given a trained model with dropout f_{nn}. To derive the uncertainty for one sample x we collect the predictions of T inferences with different dropout masks. Here $f_{nn}^{d_i}$ represents the model with dropout mask d_i. So we obtain a sample of the possible model outputs for sample x as

$$f_{nn}^{d_0}(x), ..., f_{nn}^{d_T}(x) \tag{1}$$

We obtain an ensemble prediction by computing the mean and the variance of this sample. The prediction is the mean of the model's posterior distribution for this sample and the estimated uncertainty of the model regarding x.

$$Predictive\,Posterior\,Mean,\, p = \frac{1}{T}\sum_{i=0}^{T} f_{nn}^{d_i}(x) \tag{2}$$

$$Uncertainty,\, c = \frac{1}{T}\sum_{i=0}^{T}[f_{nn}^{d_i}(x) - p]^2 \tag{3}$$

The dropout model is not modified, only the outcomes of the stochastic forward passes are collected. Through this technique, the predictive mean and model uncertainties are evaluated. As a result, existing dropout trained models can have the data applied to them.

IV. Experiments

A. Classification Models

1) BERT [4]: This model is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. As opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of words at once. Therefore it is considered bidirectional, though it would be more accurate to say that it's non-directional. This characteristic allows the model to learn the context of a word based on all of its surroundings (left and right of the word).

2) XLNet [13]: One major issue with BERT is essentially its pre-training objective on masked sequences i.e the Denoising Autoencoding objective. XLNet is a generalized autoregressive pretraining method that enables learning bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order and overcomes the limitations of BERT thanks to its autoregressive formulation. It doesn't have the denoising of inputs as in the autoencoding objective and removes the unidirectionality from a traditional autoregressive objective.

Each model of our network ends with an MCD layer (at a 35% rate). This is an improvement since it removes all evidence of dependency between the neurons and permits us to quantify uncertainty. The procedure includes randomly setting neuron outputs to zero at a set rate, simplifying the model even further.

B. Experimental Setup and Training Details

The training and testing methods for this experiment are developed using Python libraries such as Tensorflow, and Keras. An NVIDIA RTX 3080Ti GPU with 34,1 TeraFLOPS of performance is used to train and assess the models.

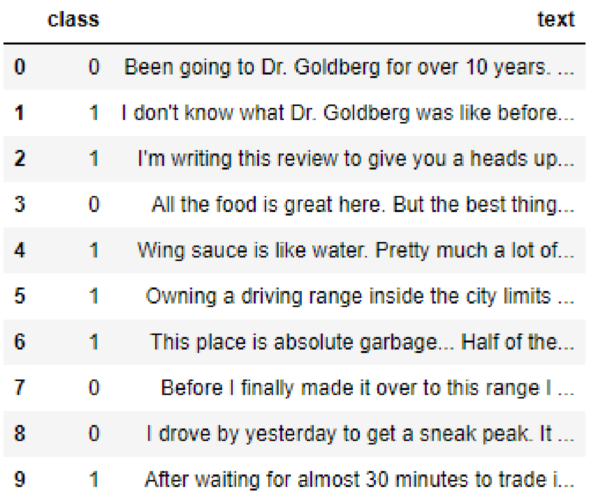
TABLE I
Performance of MCD-Based BERT, and XLNet Models

Model	Accuracy	Precision	Recall	F1 Score
BERT-MCD	87.05%	87.32%	87.13%	87.22%
XLNet-MCD	84.92%	85.24%	85.05%	85.13%

In our experiment, the dropout rates are adjusted to 30% for comparative purposes, and the models are trained with 5 epochs with and without the usage of MCD. In every experiment model, we set the learning rate at 0.0000006. The number of parameters for BERT, and XLNet are 109 Million, and 110 Million respectively. The batch size for all tests is set to 64.

C. Dataset

The Yelp reviews polarity dataset [14] is constructed by considering stars 1 and 2 negative, and 3 and 4 positive. For each polarity 280,000 training samples and 19,000 testing samples are take randomly. In total there are 560,000 training samples and 38,000 testing samples. Negative polarity is class 1, and positive class 2. In our experiment, we convert the positive class to 0. In Fig. 1 we can see the first 10 samples of the processed dataset.



	class	text
0	0	Been going to Dr. Goldberg for over 10 years. ...
1	1	I don't know what Dr. Goldberg was like before...
2	1	I'm writing this review to give you a heads up...
3	0	All the food is great here. But the best thing...
4	1	Wing sauce is like water. Pretty much a lot of...
5	1	Owning a driving range inside the city limits ...
6	1	This place is absolute garbage... Half of the...
7	0	Before I finally made it over to this range I ...
8	0	I drove by yesterday to get a sneak peak. It ...
9	1	After waiting for almost 30 minutes to trade i...

Fig. 1. First 10 Samples of the Yelp Review Polarity Dataset

D. Performance Evaluation

The model's quality is evaluated using performance assessment measures once it has completed any picture classification task. Performance evaluation metrics such as Accuracy, Recall, and Precision are used in quantitative assessments to gauge performance.

For the performance evaluation, we use baseline transformer models (BERT, XLNet), and MCD-based transformer models with a dropout rate of 35% (BERT-MCD, XLNet-MCD).

E. Measuring Uncertainty

Using MCD embedded models with 35% dropout rate, to determine the distribution of predictions, we utilize 500 test

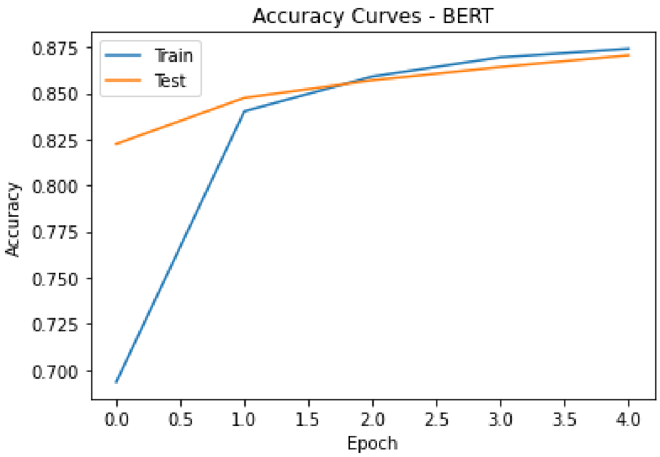


Fig. 2. Train vs Test Accuracy Curve of BERT-MCD

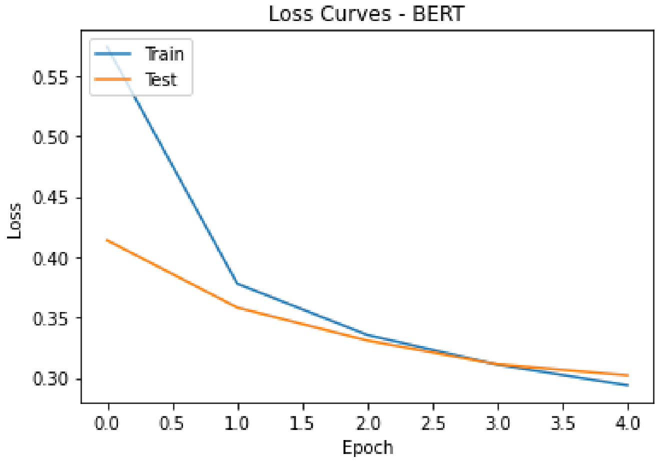


Fig. 3. Train vs Test Loss Curve of BERT-MCD

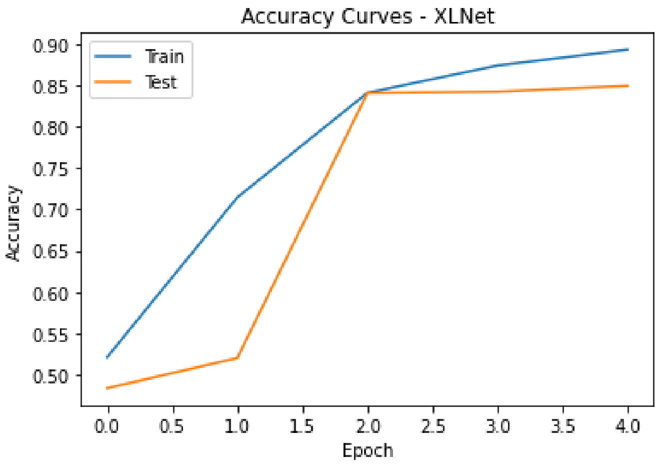


Fig. 4. Train vs Test Accuracy Curve of XLNet-MCD

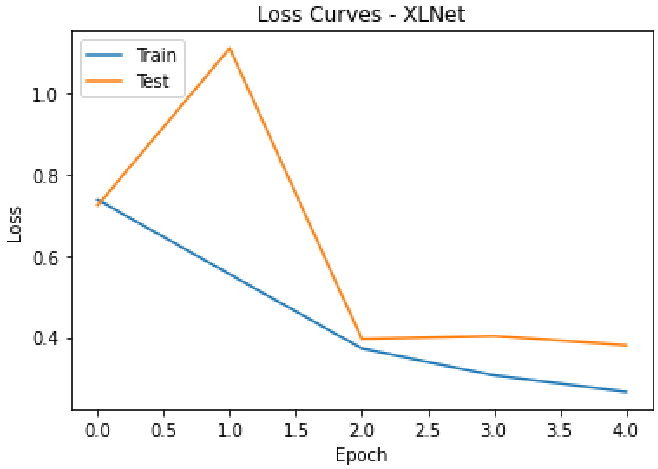


Fig. 5. Train vs Test Loss Curve of XLNet-MCD

TABLE II
UNCERTAINTY ESTIMATION OF 2 RANDOM TEST SAMPLES

Model	Predicted Class	True Class	Entropy
BERT-MCD	0	0	0.12
XLNet-MCD	0	0	0.73
BERT-MCD	1	1	0.46
XLNet-MCD	0	1	0.92

samples and predict each sample 500 times (Monte Carlo Sampling). This is needed to measure the uncertainty from the predicted class-wise softmax score distribution of the 500 test samples.

Now, we locate the most uncertain cases. This will be beneficial for comprehending our dataset or identifying problematic areas of the model. We select the most uncertain samples from the monte carlo prediction, using the variance of their softmax score.

The predictive entropy is utilized for evaluating the model uncertainty on a specific image. An uncertain sample is selected, and the predictive entropy relays how "surprised" the model is to see the particular image. The model is said to be sure about its prediction's accuracy if the value is "low". Similarly, a "high" value insinuates that the model is uncertain about the image.

$$Entropy,\ H \approx -\sum_{c}^{C}(\mu_c)log(\mu_c) \tag{4}$$

We calculate entropy using (4) where, $\mu_c = \frac{1}{N}\sum_n p_c{}^n$ is the class-wise mean softmax score. By computing the variance of the anticipated softmax score, we choose the images. Softmax turns the actual values into probabilities, therefore the score we get are simply probabilities. Additionally, the image indexes are sorted to locate an uncertain sample from the test set of data. We compare the uncertainty by taking a random sample from the test dataset to see how well the model performs with a new meaningful data.

V. CONCLUSION AND FUTURE WORK

In this work, we evaluated several uncertainty estimation for the state-of-the-art Transformer model BERT and XLNet model in the text classification tasks. To obtain estimates, we leverage multiple stochastic passes using the MCD. We show that by activating all dropouts in the end of the model for stochastic predictions, uncertainty can be estimated. Moreover, MCD boosts up the performance of models, by reducing the overfitting and increasing the test prediction accuracy. Our scheme can separate uncertain samples, reducing the risk factors of real world scenarios. Our future work focuses on working with automated uncertainty reasoning in text classification tasks.

References

[1] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/3ea2db50e62ceefceaf70a9d9a56a6f4-Paper.pdf

[2] R. Herbei and M. H. Wegkamp, "Classification with reject option," *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, vol. 34, no. 4, pp. 709–721, 2006. [Online]. Available: http://www.jstor.org/stable/20445230

[3] A. Shelmanov, E. Tsymbalov, D. Puzyrev, K. Fedyanin, A. Panchenko, and M. Panov, "How certain is your Transformer?" in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 1833–1840. [Online]. Available: https://aclanthology.org/2021.eacl-main.157

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018. [Online]. Available: https://arxiv.org/abs/1810.04805

[5] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," 2020. [Online]. Available: https://arxiv.org/abs/2003.10555

[6] A. Vaswani, S. Bengio, E. Brevdo, F. Chollet, A. N. Gomez, S. Gouws, L. Jones, Kaiser, N. Kalchbrenner, N. Parmar, R. Sepassi, N. Shazeer, and J. Uszkoreit, "Tensor2tensor for neural machine translation," 2018. [Online]. Available: https://arxiv.org/abs/1803.07416

[7] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," ser. ICML'16. JMLR.org, 2016, p. 1050–1059.

[8] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/9ef2ed4b7fd2c810847ffa5fa85bce38-Paper.pdf

[9] M. Teye, H. Azizpour, and K. Smith, "Bayesian uncertainty estimation for batch normalized deep networks," 2018. [Online]. Available: https://arxiv.org/abs/1802.06455

[10] Y. Hu and L. Khan, "Uncertainty-aware reliable text classification," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery amp; Data Mining*, ser. KDD '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 628–636. [Online]. Available: https://doi.org/10.1145/3447548.3467382

[11] A. Vazhentsev, G. Kuzmin, A. Shelmanov, A. Tsvigun, E. Tsymbalov, K. Fedyanin, M. Panov, A. Panchenko, G. Gusev, M. Burtsev, M. Avetisian, and L. Zhukov, "Uncertainty estimation of transformer predictions for misclassification detection," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8237–8252. [Online]. Available: https://aclanthology.org/2022.acl-long.566

[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[13] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1906.08237

[14] X. Zhang, J. Zhao, and Y. LeCun, "Character-level Convolutional Networks for Text Classification ," *arXiv:1509.01626 [cs]*, Sep. 2015.